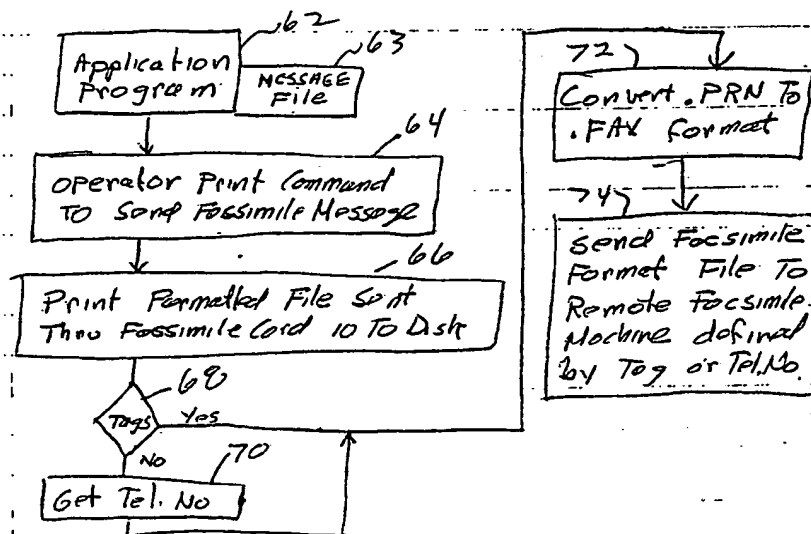




INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification ⁴: H04N 1/00, 1/32, 1/40 G06K 15/00, G06F 15/4	A1	(11) International Publication Number: WO 88/ 06824 (43) International Publication Date: 7 September 1988 (07.09.88)
(21) International Application Number: PCT/US88/00832 (22) International Filing Date: 3 March 1988 (03.03.88) (31) Priority Application Number: 021,452 (32) Priority Date: 4 March 1987 (04.03.87) (33) Priority Country: US (71) Applicant: DEST CORPORATION [US/US]; 1065 South Rogers Circle, Boca Raton, FL 33431 (US). (72) Inventors: STEVENS, Robert, K. ; 601 Goldenharbor Drive, Boca Raton, FL 33432 (US). FRIEDMAN, Gregory, A. ; 19260 Chaplecreek Drive, Boca Raton, FL 33434 (US). NAGEL, Dennis, M. ; 9 Southeast 4 Avenue, Delray Beach, FL 33444 (US). BROWN, Eric, D. ; 5220 Las Verdes Circle, Delray Beach, FL 33445 (US).	(74) Agent: BARRON, Harry, W.; 8221 Glades Road, Suite 202, Boca Raton, FL 33434-4033 (US). (81) Designated States: AT (European patent), AU, BE (European patent), BR, CH (European patent), DE (European patent), DK, FI, FR (European patent), GB (European patent), IT (European patent), JP, KR, LU (European patent), NL (European patent), SE (European patent). Published <i>With international search report. Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.</i>	

(54) Title: FACSIMILE CARD FOR A PERSONAL COMPUTER**(57) Abstract**

A facsimile card (10) for insertion into a personal computer (35), together with proper facsimile programming, allows the computer (35) to function as a facsimile device for transmitting and receiving documents. The facsimile program operates in the background of an application program (62) and includes methods for converting the facsimile data into printer graphic data (Fig. 7) when receiving facsimile message. To permit transmitting a facsimile message (63) directly from the application program (62), the facsimile program includes a method for intercepting print data (66), converting such intercepted data into facsimile data (72), and transmitting the converted data (74) to a remote facsimile machine (61). The facsimile software includes a print spooler, and a windowing foreground application program for controlling all of the functions. Alternatively, tags in the message may replace the window data entry. The card (10) may be set as an input/output port of the computer (35), so messages are sent directly from the application program (62), without interruptions.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AT	Austria	FR	France	ML	Mali
AU	Australia	GA	Gabon	MR	Mauritania
BB	Barbados	GB	United Kingdom	MW	Malawi
BE	Belgium	HU	Hungary	NL	Netherlands
BG	Bulgaria	IT	Italy	NO	Norway
BJ	Benin	JP	Japan	RO	Romania
BR	Brazil	KP	Democratic People's Republic of Korea	SD	Sudan
CF	Central African Republic	KR	Republic of Korea	SE	Sweden
CG	Congo	LI	Liechtenstein	SN	Senegal
CH	Switzerland	LK	Sri Lanka	SU	Soviet Union
CM	Cameroon	LU	Luxembourg	TD	Chad
DE	Germany, Federal Republic of	MC	Monaco	TG	Togo
DK	Denmark	MG	Madagascar	US	United States of America
FI	Finland				

FACSIMILE CARD FOR A PERSONAL COMPUTER

This invention relates to an attachment card for a computer and more particularly to such a card for permitting the computer to perform a facsimile function.

5 In the modern world, people have become accustomed to communicating with one another through the use of facsimile devices through which documents can be transferred from one location to another location. In the past, it was necessary to generate a hard copy of such a document prior to transmitting the document over the facsimile device. A
10 further delay was required when the facsimile device was in use.

The present office typically includes personal computers which are used for many clerical functions, such as word processing. When a person desires to sent a
15 textual document to another location, the text is first typed into the computer, then printed and finally scanned and transmitted by a facsimile device. Each time the document is printed, copied, scanned, and transmitted, a certain amount of resolution is lost. While this lost
20 resolution may be acceptable for textual documents, it may not be acceptable if graphic information, such as generated by CAD/CAM techniques, were included in the document. This is particularly true, where low resolution printers are used.

25 It would be preferable to have a facsimile device which is included as a part of the computer, such as a conventional attachment card for an IBM or compatible personal computer, whereby a person could generate a document containing text and/or graphical information by
30 using the computer in its intended fashion. Then, the generated document is sent directly by the computer over a communication link, such as telephone lines, to another facsimile machine or another similarly configured computer at a remote location. This technique of using the computer
35 as a facsimile device has three major advantages. First,

the resolution of the document transmitted is not reduced by printing, copying and scanning. Second, time is saved by not having to wait to print the document, carry the document to the facsimile device and wait for the transmission. Third, by using a computer as a facsimile machine, certain of the preexisting computer hardware can be used in the facsimile function, thereby saving cost.

To be even more useful, a facsimile attachment card and its associated software used with the personal computer should allow the facsimile function to operate independently and simultaneously with a standard application program (eg a word processor, accounting etc.). As used herein, the term "foreground" is intended to mean the application program the user generally desires to be using and the term "background" refers to the collection of facsimile software functions that operate simultaneously with and independently of the normal foreground application program. This is sometimes referred to in the art as multitasking.

In order to make a transmission of a facsimile message as simple as possible, it is desirable that the message be transmitted directly from the application program without the user ever leaving the application program. In other words, once a message is typed, for example, by using a word processing application program, it is desirable that a simple command be generated within the application program to cause a facsimile message to be transmitted to a remote facsimile machine. Many application programs support the use of two or more printers to which a document may be transmitted for printing. On the other hand, most computer users only have one printer connected to the computer. By designating the facsimile attachment card, and the associated computer program, as one of the unused printer paths which may be selected by the application program, the user may easily transmit the message directly to the facsimile portion of the computer for subsequent transmission to another facsimile machine.

Even where the application program only supports a single printer, the message may be encoded or commands may be entered to determine whether the message is a facsimile message to be transmitted or a normal document intended to be printed. Other types of identifying messages may be also be included on the document such as the telephone number of the remote receiving facsimile machine. By including the necessary information, which will vary from message to message, the user of the application program will be interrupted as little as possible when sending a facsimile directly from the application program.

In accordance of one aspect of this invention there is provided a method of controlling the transmission of a facsimile message directly from a computer application program which electronically stores and transmits to a designated printer encoded data manifesting the message. The method comprises the steps of designating a first facsimile machine, associated with the computer, as a printer to which the application can transmit the encoded data and, upon operator command to transmit a facsimile message given while using the application program, intercepting the electronically encoded data transmitted by the application program, converting the intercepted encoded data to facsimile formatted data and transmitting the facsimile formatted data to a second facsimile machine remote from the computer.

One preferred embodiment of the subject invention is hereafter described with specific reference being made to the following Figures, in which:

Figure 1 is a block diagram of the circuitry forming the facsimile attachment card of the subject invention;

Figure 2 is a diagram showing the manner in which Figures 2A, 2B, 2C and 2D fit together;

Figures 2A, 2B, 2C and 2D are a more detailed diagram of the block diagram shown in Figure 1;

Figure 3 is a detailed circuit diagram of a conventional data modem which can be used with the subject invention;

Figure 4 is a block diagram of the software used to operate the personal computer to allow the computer having the attachment cards of Figure 2 and 3 inserted therein;

Figure 5 shows the manner in which the facsimile attachment card may be coupled as a printer output of the computer;

Figure 6 is a flow diagram showing how the facsimile program associated with the facsimile attachment card operates when transmitting a message by facsimile communication; and

Figure 7 is a flow diagram showing how the facsimile program associated with the facsimile attachment card operates when receiving a message by facsimile communication.

Referring now to Figure 1, a block diagram of the facsimile attachment card 10 of the subject invention is shown. Card 10 is adapted to be inserted into the bus 12 of a conventional IBM or compatible personal computer 35 in a known manner. Card 10 includes the conventional DMA and I/O Request signals found on bus 12. In addition, card 10 has an internal bus 14 to which is coupled a microprocessor 16 (MPU), read only memory (ROM) 18, random access memory (RAM) 20 and optionally, a data service device (DSD) 22, such as an encryption chip of conventional design.

Also coupled to the internal bus 14 is a dual tone multi-frequency (DTMF) circuit 24 for generating what is commonly known as Touch Tone signals, which is controlled by processor 16 control signals. Also coupled to the internal bus 14 is a facsimile modem (FMO) 26, which is controlled by the output of circuit 24. The output of facsimile modem 26 is coupled to a data access arrangement (DAA) circuit 28, which provides acoustical data over

telephone line 30. When in the receive mode, acoustical data is received by DAA circuit 28 and transmitted to facsimile modem 26.

A data modem (DMOD) 32, shown in detail in Figure 3, may be included with attachment card 10 to allow attachment card 10 to function as both a facsimile device and a conventional 300/1200 baud asynchronous data modem. If data modem 32 is used, then a pair of switches 34 and 36, are used to switch the incoming and outgoing signals to and from circuit 28 between one of the two modems 26 and 32. Each of modems 26 and 32 include off hook signals F and D respectively to allow the modems 26 and 32 to take the telephone line off hook through OR gate 38 and DAA circuit 28. A speaker 40, driven by amplifier 42, is coupled to the acoustical input line between circuit 28 and switch 34.

Microprocessor 16 and the personal computer 35 communicate with one another through interface (I/F) circuit 44 and the personal computer bus 12. This path is used by the associated facsimile software in personal computer 35 to control the operation of the attachment card 10, in the manner shown in Figure 4 and described in more detail hereafter.

As with any facsimile device, a document scanner is required to scan existing documents for subsequent transmission by the facsimile device. Such a scanner may be attached through connector 46 to a scanner controller (SCNR) 48, which is controlled by microprocessor 16 over internal bus 14, to transfer the scanned data to the memory, typically the disk of the personal computer 35 using the Direct Memory Access (DMA) device of the personal computer 35.

Lastly, a conventional serial interface (COM) circuit 50 is included in card 10 and coupled through a connector 52. Circuit 50 is under the control of personal computer 35 over bus 12 by conventional I/O and interrupt request signals. When data modem 32 is used, it may be coupled at the output of circuit 50 using connector 54.

Referring now to Figures 2A through 2D, coupled as shown in Figure 2 and to Figure 3, the detailed circuitry of the attachment card 10 is shown. Like blocks of Figure 1 and corresponding chips or groups of chips in Figures 2 and 3 have been given identical numbers. The components in Figure 2 and 3 are coupled as illustrated and may be the following components:

In Figure 2A-2D:

	U1-----	LM386N-1
10	U2-----	MC6859L
	U3-----	MC6821P
	U4-----	P27256-3
	U5-----	SN74LS224N
	U6-----	SN74LS224N
15	U7-----	SN74LS224N
	U8-----	SN74LS224N
	U9-----	SN74LS04N
	U10-----	TR96-D100-211
	U11-----	D5C060-55
20	U12-----	TC5564PL-20
	U13-----	MC6802P
	U14-----	MK5380N
	U15-----	SN74LS374N
	U16-----	SN74LS125N
25	U17-----	SN74LS74N
	U18-----	SN74LS06N
	U19-----	SN74LS125N
	U20-----	SN74LS04N
	U21-----	SN74LS138N
30	U22-----	SN74LS367N
	U23-----	SN74LS74N
	U25-----	DG303ACJ
	U26-----	INS8250N
	U27-----	SN74LS682N
35	U28-----	CH1810
	U29-----	SN74LS164N
	U32-----	SN74LS374N

U33-----SN74LS245N
 U35-----SN74LS06N
 U36-----SN74LS14N
 U37-----MC1489A
 5 U38-----MC1489A
 U39-----MC1488
 U40-----SN74LS30N

In Figure 3:

10 U3-----MC1488P
 U5-----MC1489AP
 U6-----R8203-12
 U7-----10468-11
 U9-----LM1458N
 15 U24-----SN74LS393N
 U30-----SN74LS32N
 U31-----SN74LS164N

Referring now to Figure 4, a block diagram of the
 20 software used by personal computer 35 is shown. The
 following is a description of the features of the software.

The Background Communications Manager program is
 designed to support near real-time multitasking execution
 in a Microsoft DOS personal computer environment. Using
 25 this Communications Manager program, up to six simultaneous
 processes occur to support the Facsimile application.

These processes are as follows:

- a. Facsimile data transmission (receive/transmit);
- b. Conversions between different data formats;
- 30 c. Scanning paper documents using an attached scanner
unit;
- d. Printing via the built-in print spooler;
- e. Print capture (Virtual Print Interface); and
- f. Use of a foreground application program.
- 35 other DOS application program)

The Communication Manager program is designed to operate in structured processing hierarchy:

- Level 1: (highest) - Timer and hardware service interrupts
- 5 Level 2: - Background DOS service programs
- Level 3: (lowest) - Foreground DOS applications

The Level 2 Background DOS service program layer consists of a dispatcher, which allocates the processor
10 resource to various service programs. The common thread between these service programs is that they need to perform DOS services at the same time that the foreground DOS application is running.

The level 2 background DOS service program is
15 organized in a strict priority by queues. There are 5 queues, which are

Queue 1 Facsimile Communication Service. During a facsimile document transmission process, the CPU in personal computer 35, under the control of the interrupt
20 handler program, must send data to the attachment card 10 at a rate of 256 bytes of data during each approximately 0.2 seconds. In order to do this, the DOS service program in Queue reads data from the disk in personal computer 35 and places it into a series of in-storage buffers. Then
25 the interrupt handler program feeds the buffer data to the card 10 as it requests more data.

During the facsimile document receive process, the above procedure is reversed. The card 10 generates interrupts, and the interrupt handler program reads the
30 data from the card 10 and places it into a series of in-storage buffers. Then the DOS service program writes the buffer data to a disk file. This function is handled on a round robin basis and is not preemptable by the foreground, that is, all work on this queue (e.g. the
35 transfer of each 256 bytes of data) must be complete before any other DOS service program can be exacted, including the foreground application.

Queue 2- Scheduling programs - The programs in this queue perform the scheduling for all functions in the Communication Manager. Again, this queue is not pre-emptable by the foreground, and all programs in the queue are handled in a round robin fashion. These programs do the following tasks:

- a. Inspect the Schedule File to see if there is work to do. If so, the proper program is scheduled to do the work.
- 10 b. Build the top line facsimile data files for the top-line information found on all facsimile documents (date, time company, etc.);
- c. Schedule print spooler actions;
- d. Schedule Virtual Print Interface (described
15 hereafter) print capture; files for transmission or printing via the print spooler;
- e. Copy Virtual Print Interface print capture files to their destination files;
- f. Perform background facsimile communication proto-
20 col;
- g. Open/Close facsimile files; and
- h. Log all operations.

Queue 3-Scanner Service - This queue is handled on a round robin basis and may be preemptable by the foreground application. The programs on this queue accept the data
25 from the scanner unit and convert it to facsimile format for storage in a disk file. This queue is the highest preemptable queue. The scanner needs to operate a high rate of speed, but its performance can be degraded by
30 foreground work.

Queue 4-Print Spooler service - This queue is a round robin preemptable queue. The programs on this queue read data from disk files and print to parallel or serial printers.

Queue 5-Conversion service - This queue is the lowest priority queue and is used for the long-running conversion programs that convert data from one form to another (e.g. ASCII to facsimile). There are the following types of conversions:

- a. facsimile to printer format (many types of printers);
- b. facsimile to display format; and
- c. ASCII text file to facsimile format.

In general, only one conversion is scheduled at a time. The scheduler works on the basis of scheduling classes. A scheduling class is defined by a schedule record type in the schedule file. The schedule file is organized by scheduling class and the records in each class are maintained in time order (time of the next event).

The scheduler will choose an action and start work on it if it is time to begin that action and the class is not already busy. The following scheduling classes are implemented:

- a. Transmit/Receive facsimile document;
- b. Scan a paper document;
- c. Print spool; and
- d. Convert.

The background Communication Manager gains control via software and hardware interrupts to the operating system program for personal computer. The following interrupts are used, by the Microsoft DOS operating systems to cause the identified function to occur:

- a. Int 21 - DOS service interrupt - This interrupt is used to gain control prior to each DOS interrupt that the foreground application issues. This is the primary way of gaining control for background servicing. This interrupt is especially effective because if the foreground application is doing a DOS request, DOS is most likely not busy and the background can use DOS to do disk I/O.

b. Int 08 - Timer interrupt - This interrupt is trapped as it occurs 18 times a second. If the background has work to do and DOS is not busy, the background dispatcher is given control.

5 c. Int 09 - Keyboard interrupt - This interrupt is trapped for the purpose of monitoring for "hot keys". These are certain key sequences that the user presses to cause actions to happen in the background.

d. Int 28 - This is the un-documented DOS keyboard
10 busy interrupt. DOS issues this interrupt when it is able to accept disk I/O requests. The dispatcher is called.

e. Int 17 - The Communications Manager intercepts this printer interrupt to capture print data destined for VPI and other processing by background programs.

15 f. Ints 10, 13, 25, 26 - These interrupts are trapped for the purpose of knowing when the video(10), disk(13), DOS absolute I/O (25,26) are active. The background should not interrupt these processes if they are active.

Most background managers gain control via timer
20 interrupts. This is not a very reliable way to gain control and is certainly not effective in a real time application such as facsimile transmission. The trapping of Int 21 DOS requests and entering the background BEFORE DOS becomes busy is a key feature of the real time
25 background management.

DOS interrupt 21 function 3F is a read from file function request. The file may be specified as the character keyboard. If a foreground program reads from the keyboard in this manner, DOS waits for keyboard ready and
30 does not let the background get control. The interrupt handler software changes this interrupt request into the equivalent Int 21 function 0A request to read from the keyboard. This function lets the background get control and the real-time nature of the background is preserved.

35 Certain DOS functions spend a lot of time processing inside DOS. If this occurs, the background does not gain control and the real-time performance behavior is lost. To

prevent this, the interrupt handler software traps and modifies certain DOS calls. In particular, any reads and writes to any device that request transfers of over 512 bytes are broken into multiple 512 byte requests by the Int 21 handler. In addition, the Load and Exec requests are trapped and performed outside DOS to prevent DOS from spending too much time in one shot loading programs into memory. Programs are loaded into memory in smaller portions so that the background can maintain its real-time requirements.

The background Communication Manager can significantly affect the foreground performance if not properly designed. For example, if unchecked, the background could go off for many minutes and use all of the available resources so that the foreground computer operator would not even see the results of keystrokes. The way in which the background performance is managed is based around an newly-developed adaptive design. The background is given three gross performance modes:

- a. Light resource utilization - in this mode, the background uses very little of the total computer resource (processor and disk I/O).
- b. Heavy resource utilization - in this mode, the background uses most of the computer resource.
- c. Total resource utilization - in this mode, the background uses all of the computer resource.

One of the three modes is in effect at all times for the background. The mode is chosen by a determination the background programs make on what the foreground programs are doing. Once the mode is chosen, a set of externally tunable parameters govern the resource utilization. Thus, by setting some numbers in a file, the operator can tune the amount of resource used in a particular performance mode.

The method that the background uses to learn what the foreground is doing in order to make a choice about the proper performance mode is as follows:

a. If the user presses a key, the Light mode is entered for a tunable number of timer ticks (18 per second).

5 b. If the foreground program issues DOS interrupt 21 function calls, certain of these calls will cause the light mode to be entered. Each function code may save its own tunable number of timer ticks where the background will remain in the Light mode. For example, function calls that read keyboard status should not enter the Light mode
10 because many foreground application programs simply spin on these calls and the background would remain in Light mode when the foreground is not really busy at all.

c. If the foreground program issues Int 10 BIOS interrupts to request screen updating, the foreground must
15 be busy and the Light mode is entered, again for a tunable number of timer ticks.

d. If the foreground program issues Int 13 BIOS interrupts to request disk I/O, the foreground must be busy and the Light mode is entered, again for a tunable number
20 of timer ticks.

e. If none of the above activities are sensed, the Heavy mode is entered. The Heavy mode is also entered when the timer count down ends the Light mode.

25 f. A tunable number of timer ticks is also provided as a "deadman" timer. If no activity is detected in the foreground for this number of ticks, the Total mode is entered. Any foreground activity resets this timer.

g. The Int 28 DOS keyboard loop interrupt causes the Total mode to be entered until the background has no more
30 current work, or until a keystroke is detected. In the latter case, the Light mode is entered. In the former case, the previously active mode is entered.

h. In some cases, the foreground may be doing useful work without making any of the above requests. In this
35 case the background may falsely enter the Heavy or Total modes. If this occurs, simply pressing the Alt key every

so often covers, the background to enter the Light mode and the user perceives that the foreground application is performing much faster.

i. The design of the print spooler is also adaptive. The print spooler uses a polled design when working with parallel printers to save a very valuable interrupt resource in the personal computer. An adaptive tuning mechanism is a program utility that the user starts after starting the background communication manager. The program, called TUNEPTR, builds a test print file and sends it to the background print spooler to begin printing. While the print spooler is printing the document, the user is typing numbers into the foreground TUNEPTR application. With each number, the printer speeds up or slows down. The tuning process is complete when the user has set the minimum number that produces maximum printer performance for his particular computer and printer. This program is necessary because of the huge number of combinations of computers and printers, all of which have different performance characteristics.

The result of the above design is that the system is highly productive while remaining highly responsive to the foreground user. Left alone, the computer will gradually process background work faster and faster until all of the computer resource is spent on background work (the Total mode).

When the user simply presses a key to begin some other foreground task, the background automatically slows down to accommodate the foreground application. In a fully loaded system which is transmitting a facsimile document, scanning a piece of paper and printing a received facsimile document, the performance change is noticeable. When left alone, the printer and scanner perform very rapidly. When the user types on the keyboard, the printer and scanner noticeably slow down so that the foreground application retains its responsiveness.

In actual practice, when a user incorporates facsimile card 10 into a personal computer 35, the vast majority of messages which are transmitted using facsimile card 10 are first generated in a foreground application program used with personal computer 35. For example, such a foreground application program may be a conventional word processing program, spreadsheet program, data base program or the like. Given the previously described background operation of the facsimile programs shown in Figure 4, it is desirable that the user of computer 35 be able to transmit by facsimile the message created in the foreground application program as easily as possible. In the past, using conventional facsimile machines, it was necessary to first print the document and then to scan the document through a conventional facsimile machine. Existing facsimile cards used with personal computers operates with the same philosophy in that is still necessary to create a file of the document, leave the application program and enter the facsimile program and thereafter transmit the file (document) using the facsimile program. While this is an improvement over the use of separate facsimile machines and computers, it is preferable that the user be able to send a facsimile message without ever leaving the application program.

The Virtual Printer Interface (VPI). and its associate hardware and software facilities allow the facsimile card hardware 10 to intercept and capture printer data for other uses transparently to the printing software. Any board level product, which provides a personal computer with the capability to perform Group III facsimile operations, requires an interface program to the board. Such interface program should enable the facsimile card 10 to be used with any existing application program operating in the foreground. To accomplish this, a virtual printer interface is created by placing software in the DOS interrupt path and the BIOS interrupt path that can be used by the application program print

commands. The VPI software intercepts the application program print command and copies the data temporarily to disk.

Figure 5 conceptually illustrates the manner in which
5 personal computer 35 is coupled with facsimile card 10 in order to permit the virtual printer interface feature to occur. Personal computer 35 may have a plurality of different selectable printer paths to each of which a conventional printer or other device may be coupled. For
10 example, in the IBM personal computer or IBM compatible computers which operated under the Microsoft DOS operating system, up to three different paths may be designated, which are identified as LPT1 (PRN), LPT2, and LPT3. In coupling facsimile card 10 with personal computer 35,
15 appropriate switch means are included to select facsimile card 10 as responding to one of these printer paths. For example, facsimile card 10 may be coupled as the LPT3 printer port path. The switch means may be physically included on facsimile card 10, but is shown in Figure 5
20 as coupling means 60.

The application program used with personal computer 35 typically is capable of sending data, formatted for printing, over one or more of the printer paths, to a printer. Typically, a conventional printer is coupled to
25 the PRN path and receives the data to be printed. Where the application program is able to send data to a second printer port path, the application program selects that second printer path as the same path set by coupling means 60. Thus, when a user of personal computer 35 send data to
30 the second printer port, the date it is sent through coupling means 60 to facsimile card 10. Facsimile card 10 then stores the received data back in the disk of computer 35 over the interface 12.

End of file detection is determined by waiting a user specified amount of time for no characters to arrive at the interface. For example, 5 seconds may be used. Alternatively, an @@ENDDOC@@ tag, described hereafter, may be used.

5 Where the application program only supports one printer port and when end of file is determined, the VPI software opens a window on the user's application screen and asks the user for a determination of the file's status. Three actions are allowed:

10 1) enter a phone number of a destination facsimile machine which causes the saved file to become a facsimile message;

 2) enter a valid DOS path and file name, which causes the saved file to become permanent; or

15 3) enter the tag @@SPOOL@@, which causes the file to be spooled to the background print spooler and printed on the printer normally used with personal computer 35.

 This strategy allows any program which can print using the DOS print interface to automatically create
20 files for transmission to the facsimile card 10, permanent disk storage or printing on a background spooled printer. Without this process, there is no method available for foreground application programs to gain access to the DOS Print Command and the background print
25 spooler.

 The technique described above will work with conventional programs that print through the DOS or BIOS interfaces. However, programs that print directly to the hardware printer port will not work with the above
30 method.

 The method described above depends upon an interactive window which is opened on the user's application screen to acquire the disposition information for the file. An alternative is to have the user generate
35 tags which contain the disposition information. The VPI

software, described above, scans the incoming input data to find such tags and then uses the tag data to determine the disposition of the file.

Three tags may be used in place of the window. These are:

- 5 @@TELNO (telephone number data)@@ - which can contain any valid telephone number data including but not limited to the special access code information used by long distance alternative companies.
- 10 @@FILE (DOS Path and Filename)@@ - which can contain any valid file name that can be used to permanently store the file.
- @@SPOOL@@ - which can be used to indicate that the file is to be routed to the background printer spooler.
- 15

A fourth tag is also available, which is:

- @@ENDDOC@@ - which signifies the end of the document described to be sent.

Where the application foreground program includes for example, a mail merge capability in a word processor, which permits the user to send customized documents to a distribution list, the tags may be utilized with the mail merge capability. For example, one may assign one of the mail merge labels as part of the @@TELNO(telephone number data)@@ label. Specifically, the data portion of the @@TELNO tag would be a mail merge label. This would permit the document to be customized for a plurality of phone numbers, as well as the names and address typically used with a mail merge function.

30 In view of the above, it is seen that the facsimile document can be sent directly from the foreground application program without ever leaving that program. In the simplest case, where the foreground application program supports multiple printers, the normal printer is selected for documents desired to be printed and the facsimile card 35 10 is selected for documents desired to be sent by

facsimile to another facsimile machine. Where the application program only supports one printer port, the facsimile card 10 is selected as the printer port and a print spooling technique is included with the software associated with the background facsimile program which can be selected to cause the message to be printed on the printer associated with personal computer 35. The print spooler then schedules the use of the printer associated with the personal computer 35 by scheduling the printing of received facsimile message and print commands from the application program.

Referring now to Figure 6, a flow diagram is shown which illustrates the manner in which a facsimile message may be sent by the user of personal computer 35 without ever leaving the application program used to create the message. First block 62 indicates that an application program is used to create a message, which is stored in a message file 63, typically found in the hard disk associated with personal computer 35. The message may be a simple letter, a long document or a combination of graphic and textural material, depending on the capabilities of the application program.

After the message has been completed, block 64 indicates that the operator issues a print command designed to send a message by facsimile. This may be done by issuing a print command to the second port which is coupled through coupling means 60 shown in Figure 5, to facsimile card 10 or where the application program is capable of supporting only one printer port path, a message sent to that path. Next block 66 indicates that the application program 62 sends a print formatted file through facsimile card 10 to be stored in the disk of personal computer 35.

Next, according to block 68 a determination is made whether any tags are included in the message. If so, they are identified and the instruction of the tags are followed. For example, if a facsimile message is to be sent, the @@TELNO (number)@@ message may be typed on the

document. This automatically causes the message to be sent to the identified telephone number. If only a single printer path is available for application program 62 to utilize, the @@SPOOL@@ tag is used to indicate that the message is to be printed on the printer, rather than sent as a facsimile. If at block 68 it is determined that no tags are present, then according to block 70 a window is inserted into the screen requesting the necessary information.

10 After the desired information is entered, whether by tags through block 68 or by entry according to the information requested from the window at block 70, block 72 converts the printer formatted file to a facsimile formatted file in the event a facsimile command has been
15 issued. If it merely were desired to print the message, this function then occurs and is not indicated in Figure 6.

Finally, according to block 74, the facsimile formatted file is sent to the remote facsimile device defined by the telephone number in either the tags or the
20 window entry. This may be scheduled to occur as soon as the conversion from block 72 is finished or at any other time, such as when telephone rates are lower.

When a facsimile message is received, the facsimile attachment card 10 is responsible for decoding the
25 facsimile document data (encoded as defined in the CCITT standards for Group III Document Facsimile Transmission) and converting it into data which can be directly spooled to a graphics printer. The following type of conversions take place based on the type of printer associated with
30 personal computer 35:

8-Pin 2-Pass Printer Conversion

8-Pin 1-Pass Printer Conversion

150 DPI Printer Conversion

300 DPI Printer Conversion

16-Pin Printer Conversion

24-Pin Printer Conversion

The received facsimile document data is stored encoded as a file. This file is opened and processed by the facsimile decode routine. The facsimile decode routine decodes the facsimile run-length codes into 1728 pixel data blocks (hereafter referred to as scan lines), and passes this decoded data to the print conversion routines.

The print conversion routines use a common block of memory to perform the print conversion. The shared memory is made possible by the Communications Manager program, which guarantees that only one print conversion routine may be executing at any one time. Each print conversion routine divides the block of memory into two buffers, the input buffer (hereafter referred to as the Pixel Buffer) and the output buffer (hereafter referred to as the Print Buffer). A print conversion algorithm is used to convert the decoded facsimile data from the Pixel Buffer to printable bit-image graphics and printer-specific escape sequences in the Print Buffers. A Write Buffer may also be implemented to buffer the Print Buffer

The print conversion algorithm for each print conversion routine is such that, when the Print Buffer Data is spooled to the printer, the original facsimile document is reproduced on the printer, with the horizontal and vertical aspect ratios as correct as possible. Thus, the original size of the facsimile document is approximated, and prints on a single 8.5 X 11" page. One exception is made to preserve the vertical aspect ratio in the case of the 8-Pin 1-Pass Printer Conversion routine. The resolution capabilities of an 8-pin printer making one print head pass per line are limited. To provide a higher resolution on such a printer, the 8-Pin 1-Pass Printer Conversion routine allows for the printing of a single facsimile page to overflow onto a second 8.5 X 11" printer page.

The aim of the print conversion algorithms is to reproduce the original facsimile document as accurately as possible, both in size and in resolution. The resolution of printed data is directly dependent on the number of pins in the print head of a particular printer. In most cases, the printable resolution is less than the 204 horizontal pixels per inch by 97 pixels per inch resolution of the decoded facsimile data in standard resolution. Therefore, each printer type supported by the attachment card 10 requires a unique printer conversion algorithm to convert the Pixel Buffer data to the Print Buffer data with a minimum loss of information.

The print conversion algorithms concentrate on preserving the vertical resolution more so than the horizontal resolution, for two major reasons. First, the high horizontal resolution of facsimile data permits a large amount of thinning (discarding of horizontal pixels at regular intervals) before the horizontal resolution approaches the vertical resolution. Second, the primary and most significant information in facsimile documents are alphanumeric characters, which are more sensitive to loss of vertical information than to loss of horizontal information. This can be attributed to the relatively thin horizontal critical parts of alphanumeric characters, such as the cross in a 't' or the horizontal lines in an 'e' and an 'A', whereas most vertical strokes in alphanumeric letters are bolder and rounded (incorporating some horizontal information. However, if a character is reduced to strokes of one pixel width in the original facsimile document, any thinning, vertical or horizontal, will cause a critical loss of information. Such characters are too small to be commonly encountered in a facsimile document.

The print conversion algorithms attempt to perform a direct mapping from decoded facsimile data to 1) the maximum printable resolution of the printer, if it is less than the 204 X 97 dots per inch (DPI) of facsimile data, or

2) the nearest printable resolution of the printer exceeding the resolution of facsimile data. This applies to both horizontal and vertical conversion.

When the maximum printable resolution (MPR) of the printer is greater than the facsimile data resolution (FDR), the following strategies are applied to convert the Pixel Buffer data to Print Buffer data, in both the horizontal and vertical directions:

A. If the MPR approximates an integral multiple N of the FDR, each pixel of Pixel Buffer Data is duplicated N times in the Print Buffer.

B. Otherwise, the pixels of Pixel Buffer data are uniformly distributed in the Print Buffer data, which leaves undefined gaps of pixels in the Print Buffer. If the MPR is greater than twice the FDR, each pixel is repeated a corresponding integral number of times. This leaves only solitary undefined pixels in the Print Buffer. These remaining pixels are defined to be the AND result of their two immediate neighbors.

Strategy A provides a direct mapping from Pixel Buffer space to Print Buffer space, maintaining the appearance of the original facsimile data. For conversions not falling under this category (in the case where $MPR > FDR$), a decision process is needed to determine the value of the undefined pixels. A true direct mapping is not feasible as the calculation of fractional pixel values would require arithmetic or multiplicative calculations for each bit value, which could increase the time of conversion a thousandfold.

Strategy B provides a fast conversion criterion for assigning value to the undefined pixels, and avoids a loss of information in a high percentage of facsimile documents. Since critical facsimile information is usually black text on a white background, where white pixels compose the large percentage of space, black pixels can be considered significant information. The typical decision process would be either to uniformly discard pixels until the

24

condition for Strategy A is met, or to assign undefined pixels the value 0 (a value of 1 would be assigning meaning to white space). In fact, by assigning undefined pixels the value 0 in the case where,

5

$$2 * FDR > MPR > FDR$$

a significant decay in the appearance of alphanumeric characters occurs. Although all information from the original facsimile is preserved, superfluous white space is introduced into the characters - as much as 50% of the original information becomes interspersed white space.

To exemplify how Strategy B maintains the continuity of significant information, consider mapping the pixel string M onto pixel space N:

20 M: 0 0 1 0 1 0 0 1 1 1 0
N: 0 0 _ 1 0 _ 1 0 _ 0 1 _ 1 1 _ 0
Q: 0 0 0 1 0 0 1 0 0 0 0 1 1 0 1 0
P: 0 1 0 1 1 0 1 1 0 0 1 1 1 1 1 0
25 Q: 0 0 0 1 0 0 1 0 0 0 0 1 1 1 1 0

Pixel string N shows the 11 pixels of M uniformly distributed in 16 pixels. By assigning the undefined pixels a 0 value, a discontinuity is created in the right most string of 1's. In string P, assigning undefined pixels a 1 value created spurious information in the left most 1 pixel. Strategy B gives string Q, which has lost no information from string M and maintains the continuity of significant information. In fact, in this example, Q is a direct mapping of M onto 16 pixels.

When the facsimile data resolution (FDR) exceeds the maximum printable resolution (MPR) of the printer, some loss of information from the Print Conversion Algorithms is unavoidable, since the printer does not have the capability of reproducing the facsimile data resolution. In this case, pixels are uniformly discarded from the Pixel Buffer data to achieve the MPR. Some conditions restrain a uniform reduction of the Pixel Buffer data in the horizontal direction. To achieve horizontal thinning in a time-efficient manner, horizontal reduction is done primarily on integral sets of 8 pixels (a byte) to smaller sets of 8 pixels. In these cases, duplication of pixels at uniform intervals is done after thinning, to account for the horizontal aspect ratio.

These print conversion routines and the corresponding print conversion algorithms are not printer specific. Each print conversion routine can be used for a number of printer types. Printer dependent information, i.e., escape sequences, are stored in a configuration file and the Communications Manager dynamically reads this file when scheduling print conversions.

Referring to Figure 7, a flow diagram is shown of the sequence of events when a facsimile message is received. First, according to block 76, the facsimile message is received while the operator is performing a task using the application program 62. The facsimile program, operating in the background, then stores the received facsimile message data in the disk of computer 35, as indicated by block 78. Next, as indicated by block 80, the facsimile formatted data is converted to printable data as previously described and stored in a print buffer in the disk of computer 35, as indicated by block 82. Finally, the printable data is sent to the printer 37 using the print spooler program of the background facsimile program, as indicated by block 84. All of the processing of the received facsimile message occurs without intervention of the operator using the application

26

program 62 and the first thing the operator will notice is the printer 37 beginning to print the message.

What is claimed is:

1. A method of controlling the transmission of a facsimile message (63) directly from a computer (35) application program (62) which electronically stores and transmits to a designated printer (37), encoded data
5 manifesting said message (63), said method characterized by the steps of:

designating a first facsimile machine (10, 60), associated with said computer (35), as a printer (37) to which said application program (62) can transmit said
10 encoded data; and

upon operator command (64) to transmit a facsimile message (63) given while using said application program (62),

intercepting (66) said electronically encoded
15 data transmitted by said application program (62);

converting (72) said intercepted encoded data to facsimile formatted data; and

transmitting (74) said facsimile formatted data to a second facsimile machine (61) remote from said
20 computer (35).

2. The method according to claim 1 further characterized in that said facsimile formatted data is transmitted to said second facsimile machine (61) while said operator continues to be able to use said application program (62).

3. The method according to claim 1 further characterized in that said intercepted data is converted to said facsimile formatted data while said operator continues to be able to use said application program (62).

4. The method according to claim 1 further characterized in that said encoded data is intercepted while said operator continues to be able to use said application program (62).

5. The method according to claim 4 further characterized in that said facsimile formatted data is transmitted to said second facsimile machine (61) while said operator continues to be able to use said application program (62).

6. The method according to claim 4 further characterized in that said intercepted data is converted to said facsimile formatted data while said operator continues to be able to use said application program (62).

7. The method according to claim 6 further characterized in that said facsimile formatted data is transmitted to said second facsimile machine (61) while said operator continues to be able to use said application program (62).

8. The method according to claim 7 further characterized in that said designated printer (37) and first facsimile machine (10, 60) are different.

9. The method according to claim 7 further characterized in that said designated printer (37) and first facsimile machine (10, 60) are the same.

10. The method according to claim 1 further characterized in that said designated printer (37) and first facsimile machine (10, 60) are different.

11. The method according to claim 1 further characterized in that said designated printer (37) and first facsimile machine (10, 60) are the same.

29

12. The method according to claim 11 further characterized in that:

said method further includes controlling the reception of facsimile messages (63) from other locations;
5 and

said method further includes the step of scheduling printing functions between the received facsimile messages (63) and print commands of said application program (62).

13. The method according to claim 1 further characterized in that said second facsimile machine (61) is designated by a certain number, said method further comprising the step of inserting a tag manifesting said
5 certain number as the initial part of said message (63).

14. In a computer (35) programmed with an application program (62) used to perform a desired task and a facsimile conversion and transmission program (Fig 6) operating in a multitasking mode while said application program (62) is
5 being used to perform said desired task, a method of transmitting a facsimile message (63) while using said application program (62) characterized by the steps of:

preparing said message (63) while using said application program (62);

10 adding selected encoded information containing tags defining transmitting instructions to said message (63) using conventional characters available to said application program (62);

entering a facsimile transmit command (64) to said
15 computer (35); and

transmitting (74) said message (63) in accordance with the transmitting instructions contained in said tags.

15. The method according to claim 14 further characterized by the steps of said facsimile program (Fig 6) intercepting (66) said message (63) from said application program (62), recognizing (68) said tags and transmitting (74) 5 said message (63) in facsimile format to a remote location (61) in accordance with said instruction information contained in said tags.

16. The method according to claim 15 further characterized in that said steps of intercepting (66), recognizing (68) and transmitting (74) are performed while said step of preparing is being performed for a subsequent message (63).

17. The method according to claim 16 further characterized in that said selected encoded information includes at least one designated leading alpha-numeric character, a predefined set of alpha-numeric characters defining a function, 5 a user defined series of informational alpha-numeric characters related to said function, and at least one designated trailing alpha-numeric character.

18. The method according to claim 17 further characterized in that said predefined set of alpha-numeric characters defining a function include the telephone number of a facsimile machine (61) at said remote location.

19. The method according to claim 18 further characterized in that said predefined set of alpha-numeric characters defining a function include the telephone number of a facsimile machine (61) at said remote location.

20. The method according to claim 19 wherein said application program (62) includes a mail merge function in which selected data is entered to permit similar customized messages (63) for multiple recipients to be created further characterized in that said step of adding selected encoded information containing tags includes the step of adding said information as a part of said mail merge data.

21. A facsimile transmitting machine (10, 35 and 60) including a computer (35) controlled by an application program (62) for preparing and storing the contents of a document (63) in standard encoded format data, said computer (35) including means for selecting one of a plurality of available printer paths (PRN/LPT1, LPT2 and LPT3) to which said stored standard encoded format data is provided to permit said contents to be printed by a printer (37) affixed to one of said paths (PRN/LPT1) characterized by:

means for intercepting and converting (10, 60) said standard encoded format data into facsimile format data and for transmitting electronic signals manifesting said facsimile format data over common carrier means; and

means for coupling said means for intercepting and converting (10, 60) as said printer path (LPT3) selectable by said application program (62) whenever it is desired to send said document (63) contents to a remote facsimile machine (61) for printing.

22. The invention according to claim 21 further characterized in that said means for intercepting and converting (10, 60) includes means for reading (68) predefined encoded tags included in said document (63) defining said remote facsimile machine (61) and for deleting said tags prior to transmitting said electronic signals.

23. The invention according to claim 21 further characterized in that:

said document (63) includes predefined encoded tags therein defining said remote facsimile machine (61); and

5 said means for intercepting and converting (10, 60) includes means for reading (68) said predefined encoded tags included in said document (63) and for deleting said tags prior to transmitting said electronic signals.

24. The invention according to claim 23 further characterized in that said predefined encoded tags are at the beginning of said document (63) and include at least one designated leading alpha-numeric character, a predefined
5 set of alpha-numeric characters defining a function, a user defined series of informational alpha-numeric characters defining said remote facsimile machine (61) and at least one designated trailing alpha-numeric character.

25. The invention according to claim 21 further characterized in that said means for intercepting and converting (10, 60) includes a computer program (Fig 6) operating in a background multitasking relationship with said application
5 program (62).

26. The invention according to claim 21 further characterized in that said means for intercepting and converting (10, 60) intercepts, converts and transmits said data while said application program (62) is used to prepare and store
5 the contents of another document (63).

: 33

27. A facsimile transmitting machine (10, 35, 60) for transmitting a message (63) provided thereto to a remote facsimile machine (61) having an identification designation, said message (63) having said identification designation of said remote machine (61) included therein in a pre-formatted manner, said machine comprising:

means for scanning (68) said provided message (63) for said identification designation; and

means for transmitting (74) said message (63) to said remote machine (61) in response to at least a portion of said identification designation.

28. The invention according to claim 27 in that said machine (10, 35, 60) is further characterized by means for deleting said identification designation from said message (63).

29. The invention according to claim 28 further characterized in that said identification designation includes instructions for operating said machine (10, 35, 60).

30. The invention according to claim 29 further characterized in that said identification designation includes the telephone number of said remote facsimile machine (61).

31. The invention according to claim 30 further characterized in that said identification designation includes certain alpha-numeric characters preceding and following said telephone number.

32. The invention according to claim 31 further characterized in that the sequence of said certain alpha-numeric characters are identified by said means for scanning (68).

33. The invention according to claim 27 further characterized in that said identification designation includes the telephone number of said remote facsimile machine (61).

34. The invention according to claim 33 further characterized in that said identification designation includes certain alpha-numeric characters preceding and following said telephone number.

35. The invention according to claim 34 further characterized in that the sequence of said certain alpha-numeric characters are identified by said means for scanning (68).

36. The invention according to claim 27 further characterized in that said identification designation includes instructions for operating said machine (10, 35, 60).

37. In a computer (35, 37) system, including a printer (37) and a processor (35) programmed with an application program (62) used to perform a desired task, the results of which are printed on said printer (37), said processor (35)
5 further being programmed with a facsimile conversion and transmission program (Figs. 6 and 7) operating in a multi-tasking mode while said application program (62) is being used to perform said desired task, a method of receiving a facsimile message (63) while using said application program
10 (62) characterized by the steps of:

storing (78) said message (63) in facsimile format while using said application program (62);

converting (80) said message (63) to printable formatted data while using said application program (62); and

15 transmitting (84) said data in said printable format to said printer (37) for printing said received facsimile message (63) while using said application program (62).

38. The method according to claim 37 further characterized in that said step of converting (80) includes the step of thinning the horizontal information to a greater extent than the vertical information.

35

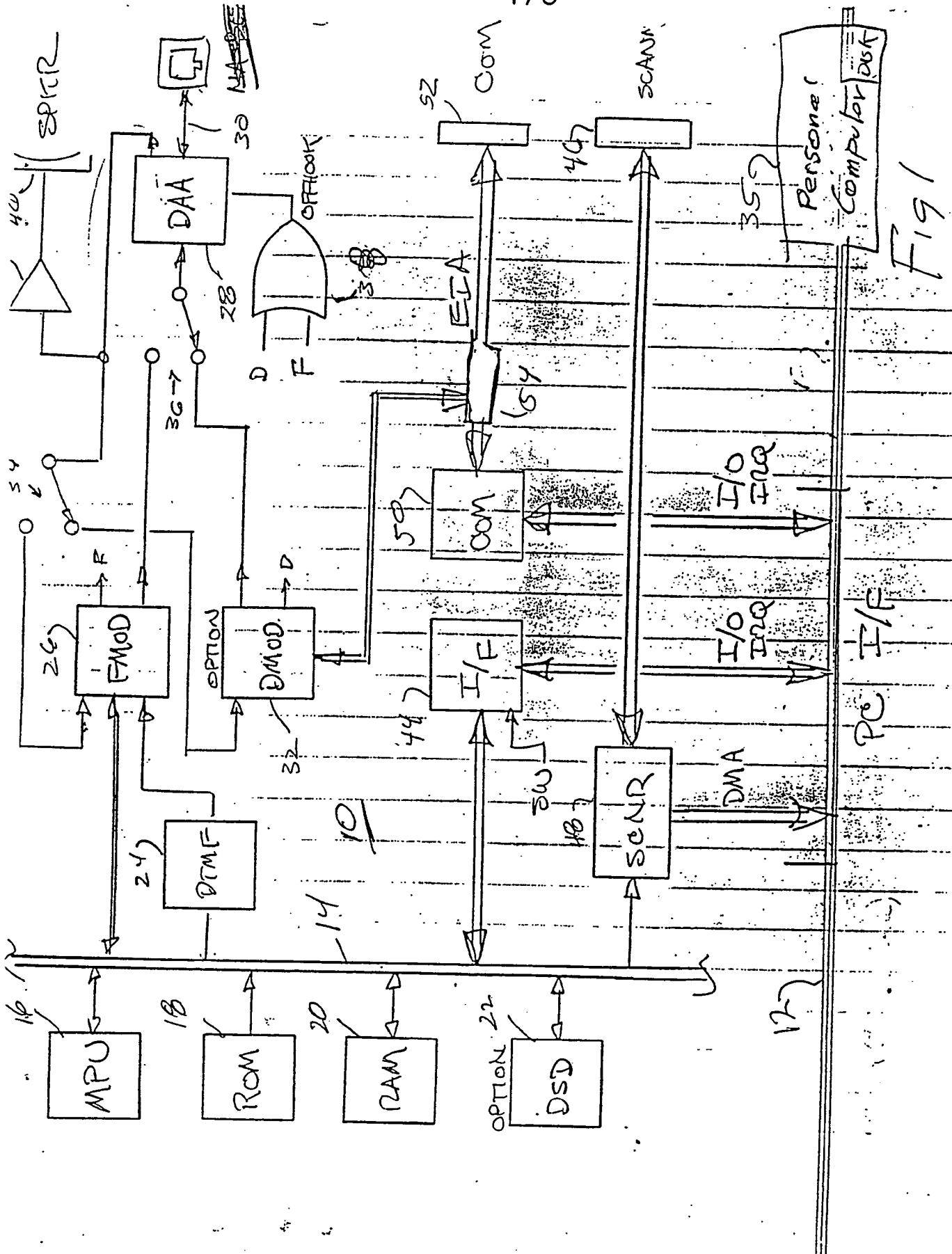
39. The method according to claim 38 wherein said processor (35) is further programmed with an operating system program which includes a printer (37) control subprogram used by said application program (62) for printing said task results, further characterized in that said step of transmitting (84) including replacing said operating system printer (37) control subprogram with a printer (37) control program which causes said printable formatted data to be printed and intercepts said application program (62) print requests for printing.

40. The method according to claim 37 wherein said processor (35) is further programmed with an operating system program which includes a printer (37) control subprogram used by said application program (62) for printing said task results, further characterized in that said step of transmitting (84) including replacing said operating system printer (37) control subprogram with a printer (37) control program which causes said printable formatted data to be printed and intercepts said application program (62) print requests for printing.

41. A facsimile receiving machine having a computer (35) programmed with an application program (62) for performing task under the control of the operator; facsimile receiving means coupled to said computer (35) and responsive to facsimile data manifesting an incoming facsimile message (63); and printer means (37) coupled to said computer (35) characterized by

said computer (35) being programmed with a facsimile control program (Fig. 7) operating in the background with respect to said application program (62) for converting said facsimile data to printable data and for controlling the printing of said facsimile message (63) by said printer means (37) in response to said printable data while the operator of said computer (35) is using said application program (62).

42. The invention according to claim 41 further characterized in that said facsimile control program (Fig 7) controls the printing of the results of performing said task during the use of said application program (62).



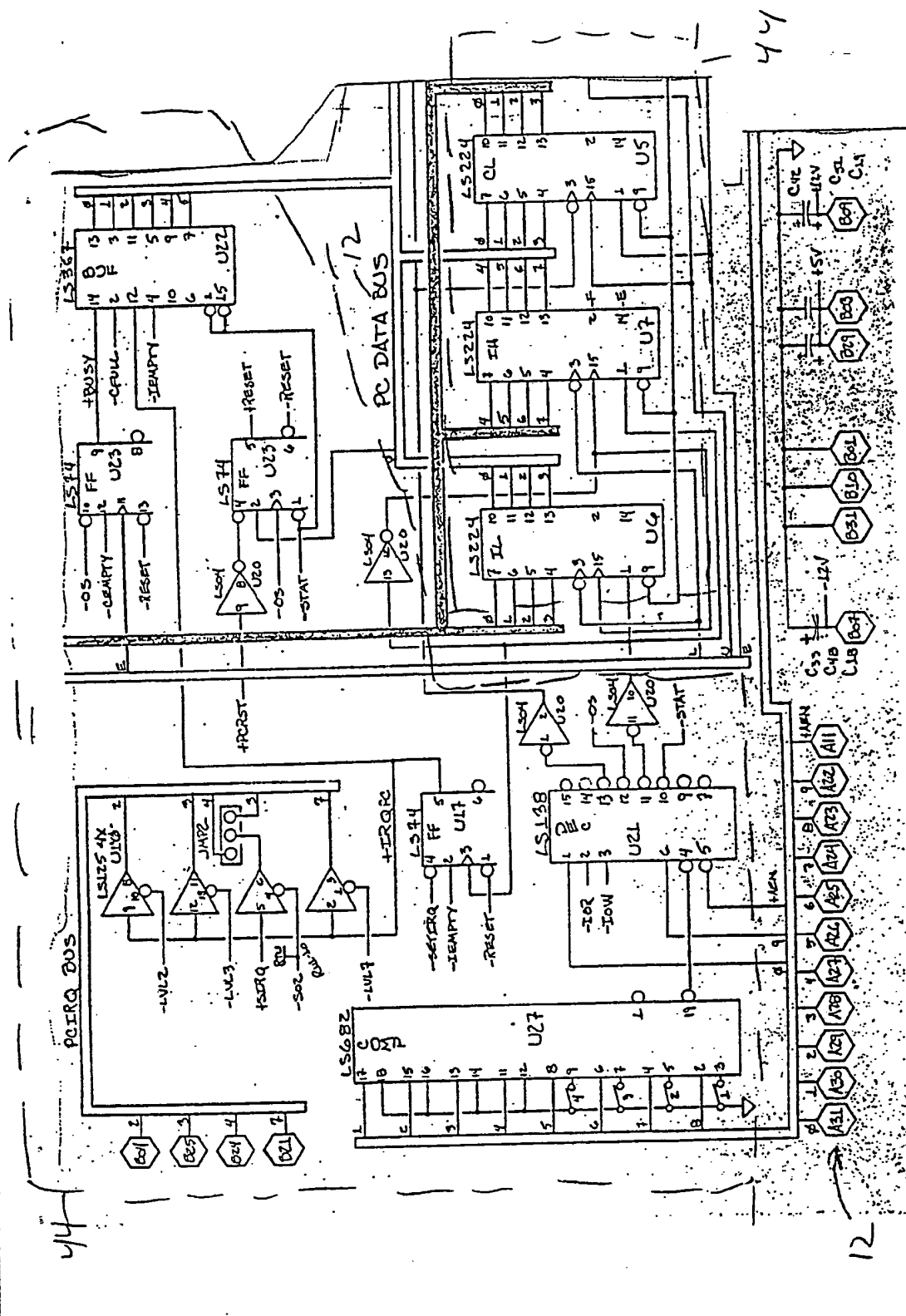
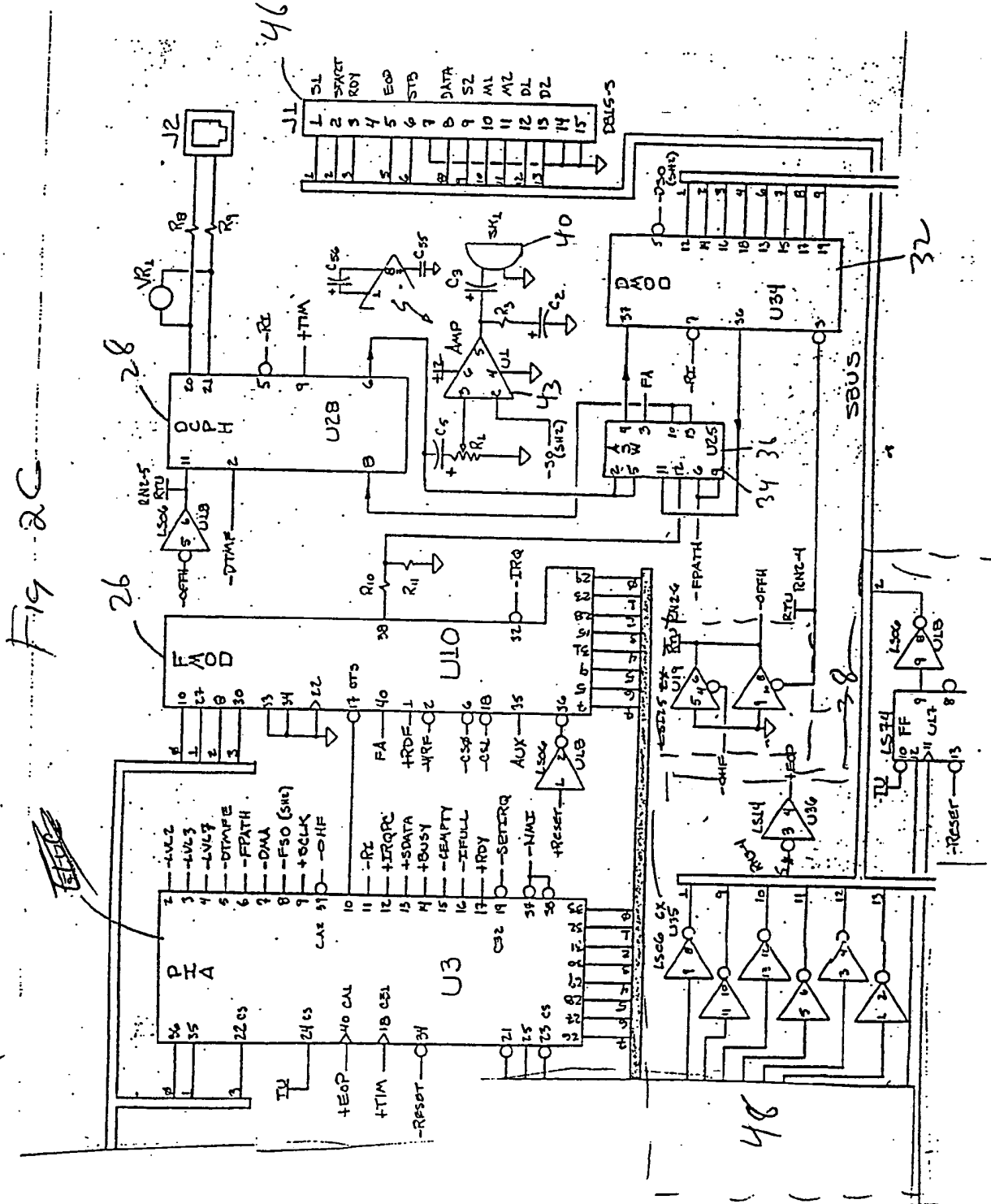
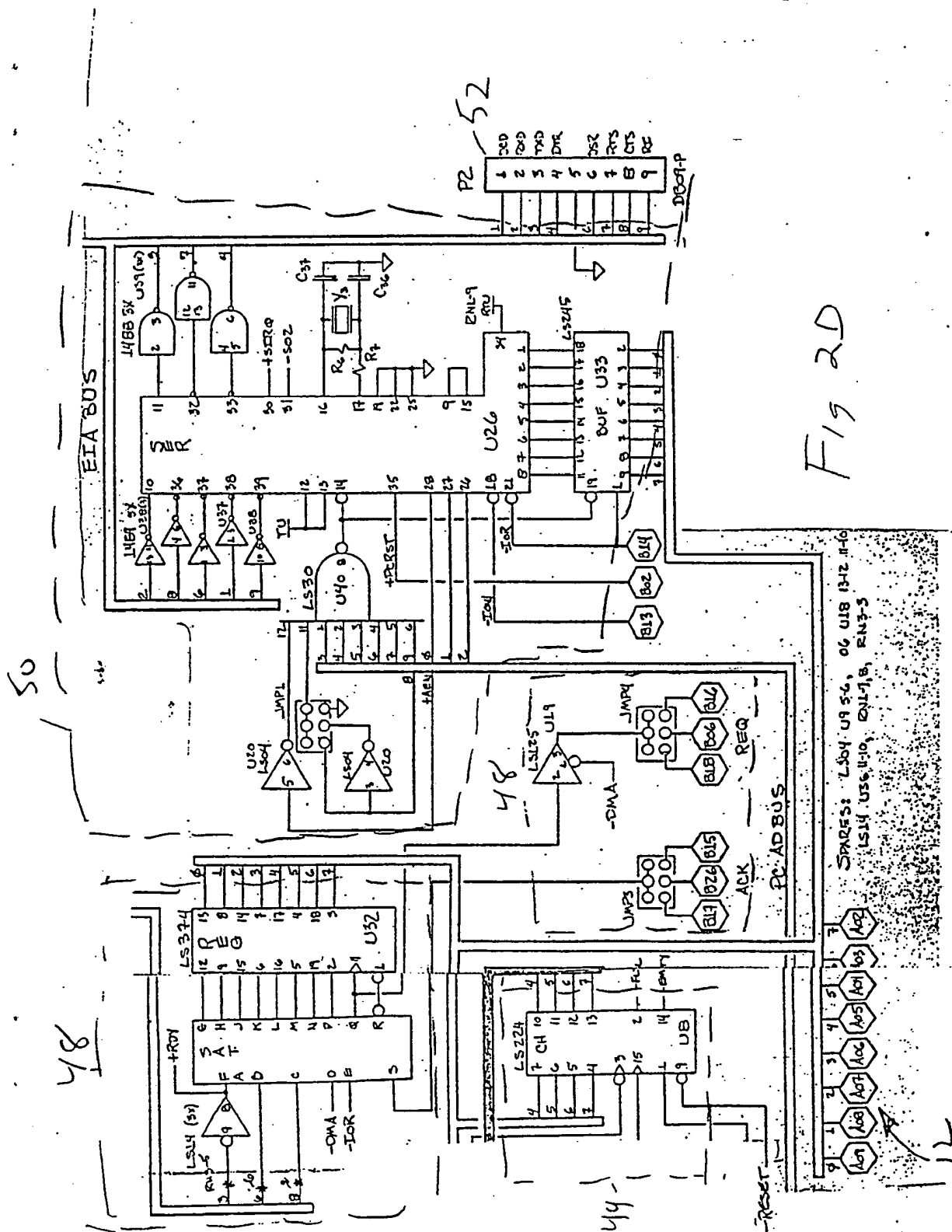


Fig. 2B

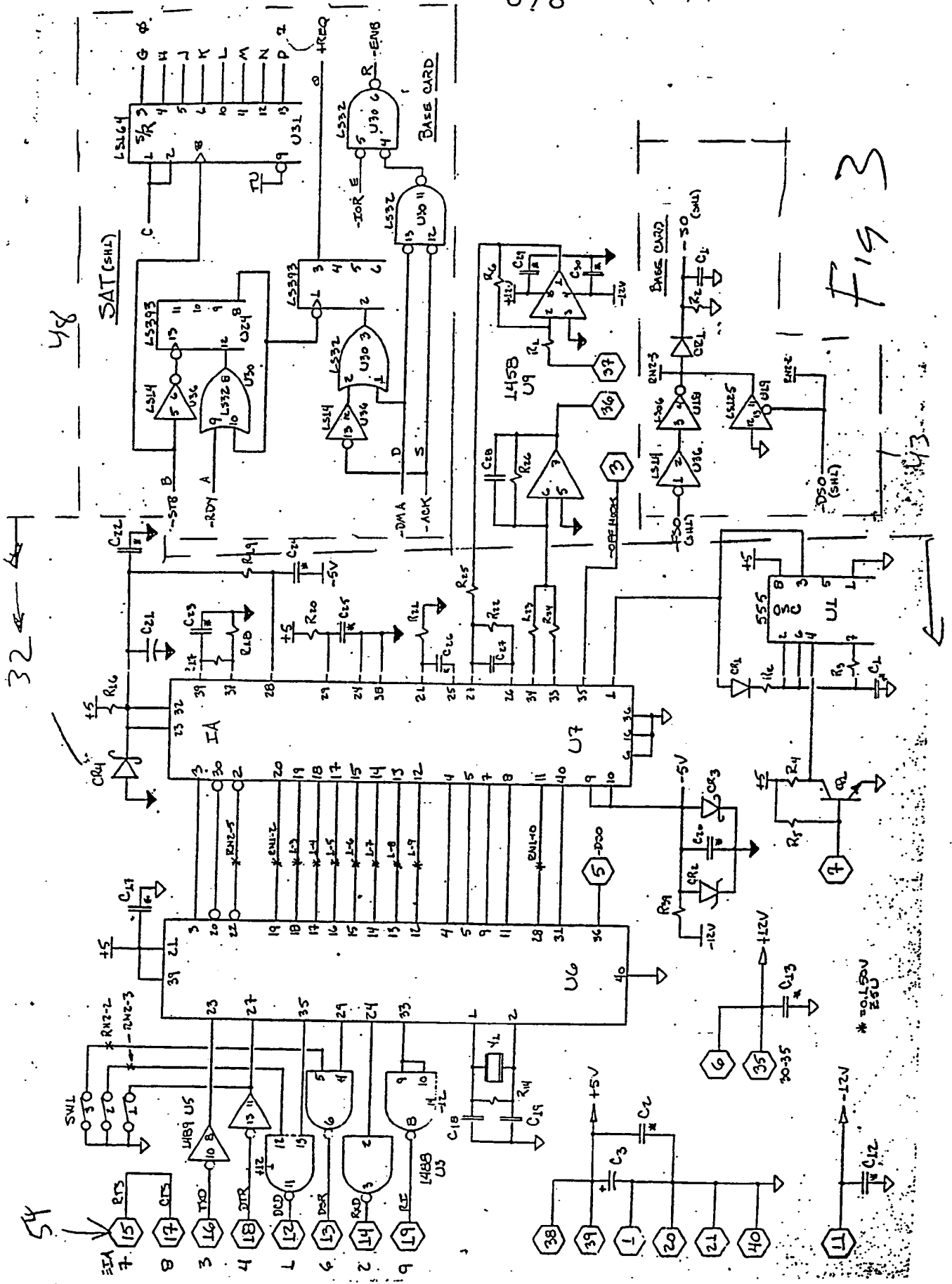
Fig 2A	Fig 2C
Fig 2B	Fig 2D

4/8





F19 2D



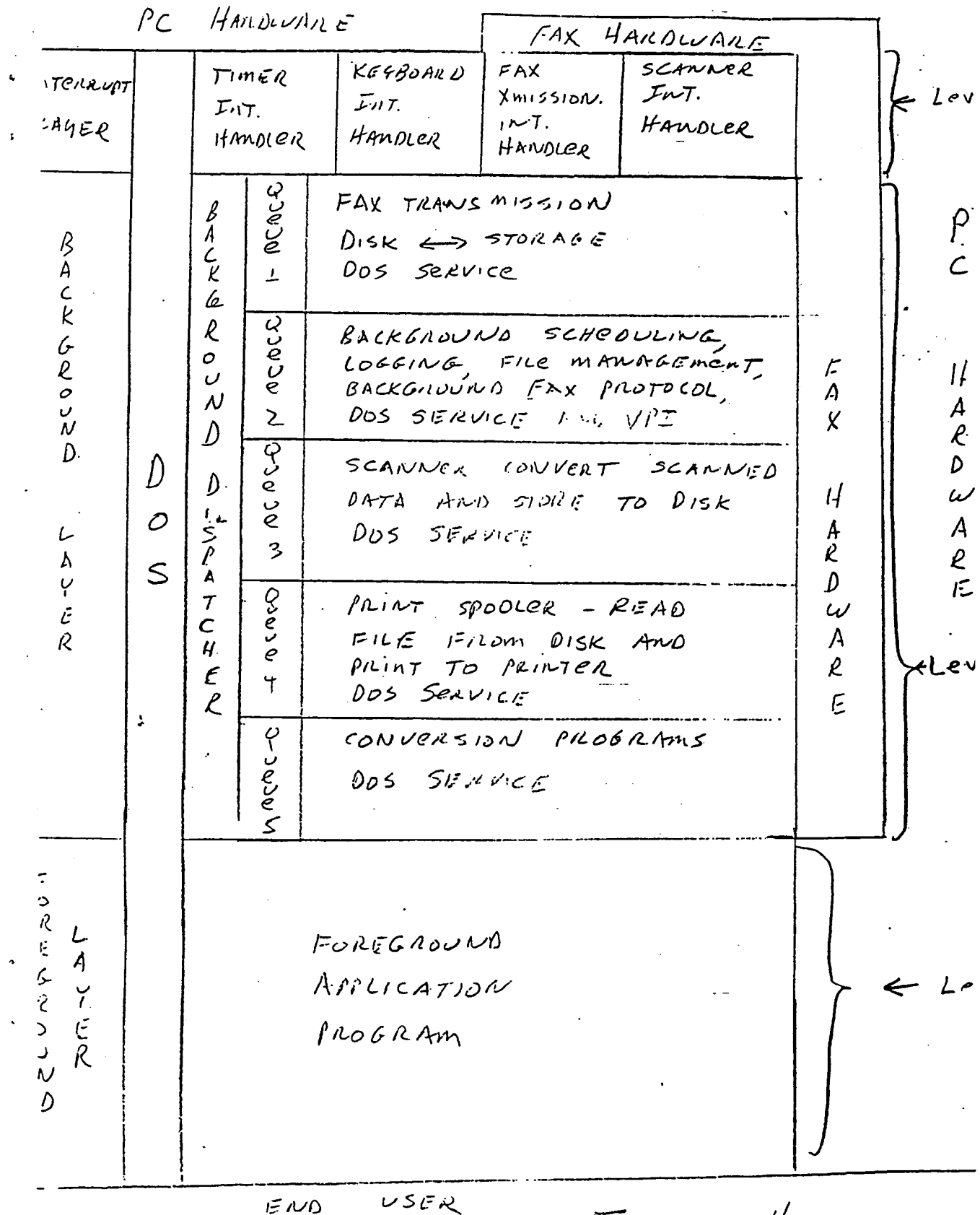
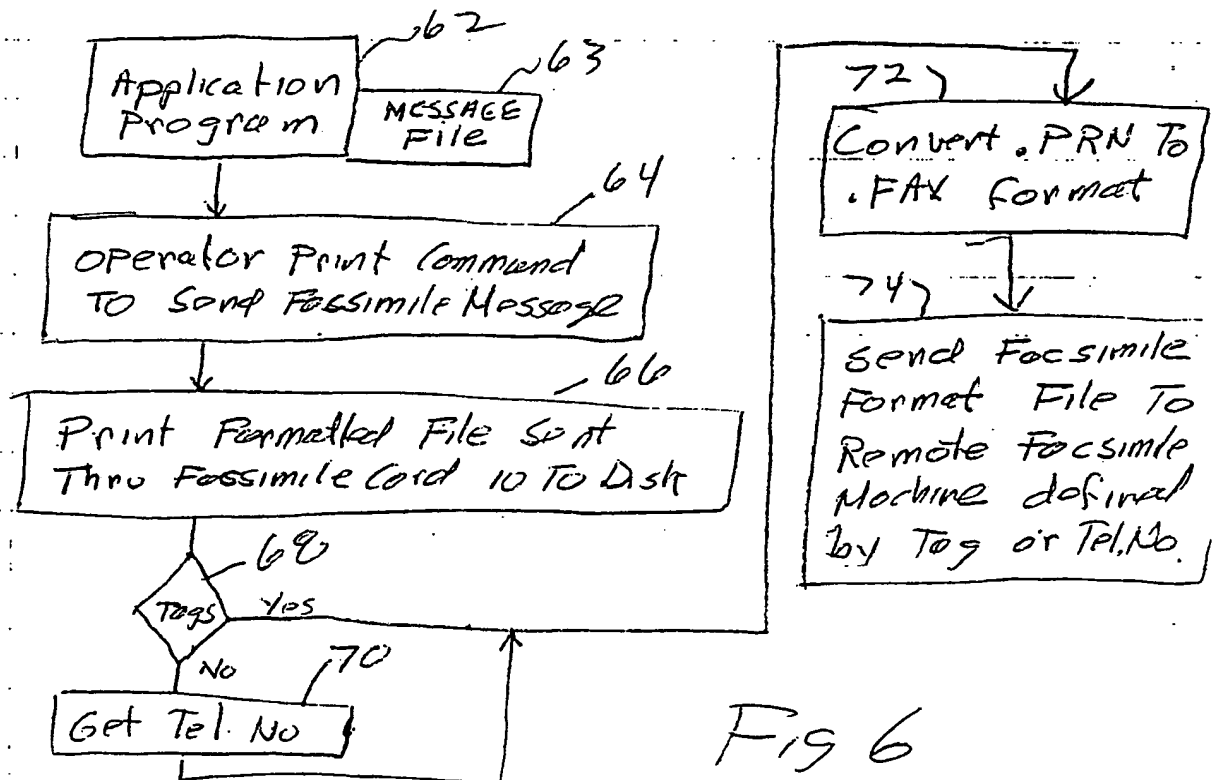
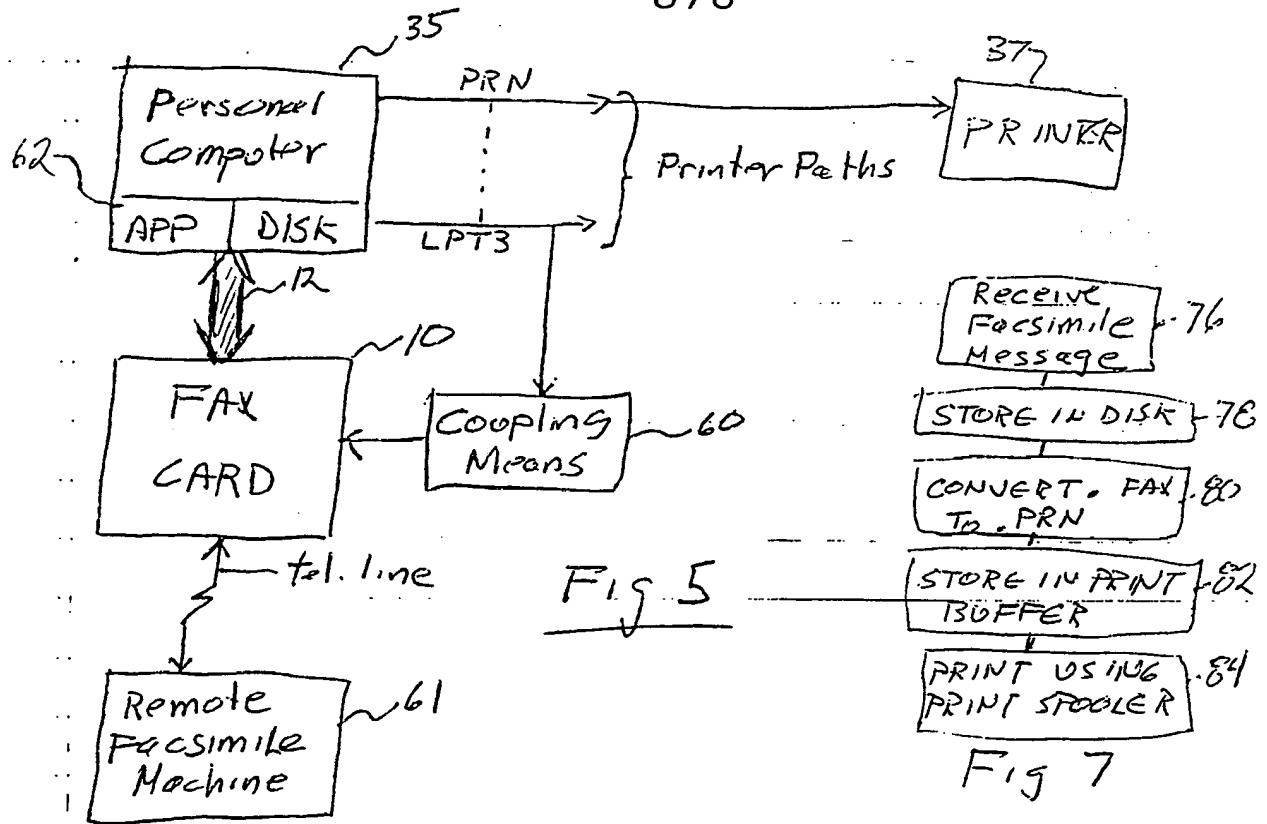
Software Block Diagram

Figure 4

8/8



INTERNATIONAL SEARCH REPORT

International Application No. PCT/US88/00832

I. CLASSIFICATION OF SUBJECT MATTER (if several classification symbols apply, indicate all) ⁶		
According to International Patent Classification (IPC) or to both National Classification and IPC INT. CL. H04n 1/00, 1/32, 1/40; G06k 15/00; G06F 15/4 US CL. 358/256, 257, 280, 286, 903; 364/514, 518		
II. FIELDS SEARCHED		
Minimum Documentation Searched ⁷		
Classification System	Classification Symbols	
US	358/256, 257, 286 364/514, 518	
Documentation Searched other than Minimum Documentation to the Extent that such Documents are Included in the Fields Searched ⁸		
III. DOCUMENTS CONSIDERED TO BE RELEVANT ⁹		
Category ¹⁰	Citation of Document, ¹¹ with indication, where appropriate, of the relevant passages ¹²	Relevant to Claim No. ¹³
Y,P	US,A, 4,652,933 Koshushi, 24 March 1987 See Col. 3, lines 45-68, Col. 4, 1-18	1-42
Y	N, Digital Research, Published August 1986, Concurrent PC DOS, See pages 2-(1-3), 4-(1-8), 6-(1-13)	1-42
Y,P	US,A, 4,713,780, Schultz et al, 15 December 1987, See Col. 3, lines 40-68, Col. 4, lines 23-32, Col. 8, lines 10-12	1-42
Y	US,A, 4,587,633, Wang et al, 06 May 1986, Col. 3, lines 1-64, Wang et al	1-42
<div style="display: flex; justify-content: space-between;"> <div style="width: 45%;"> <p>¹⁰ Special categories of cited documents:</p> <p>"A" document defining the general state of the art which is not considered to be of particular relevance</p> <p>"E" earlier document but published on or after the international filing date</p> <p>"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)</p> <p>"O" document referring to an oral disclosure, use, exhibition or other means</p> <p>"P" document published prior to the international filing date but later than the priority date claimed</p> </div> <div style="width: 45%;"> <p>"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention</p> <p>"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step</p> <p>"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.</p> <p>"&" document member of the same patent family</p> </div> </div>		
IV. CERTIFICATION		
Date of the Actual Completion of the International Search		Date of Mailing of this International Search Report
09 June 1988		29 JUN 1988
International Searching Authority		Signature of Authorized Officer
ISA/US		Edward L. Coles <i>Edward L. Coles</i>

Form PCT/ISA/210 (second sheet) (Rev.11-87)

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☐ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.